

# Convolutional Neural Network

Input : Images

Examples : Image classification

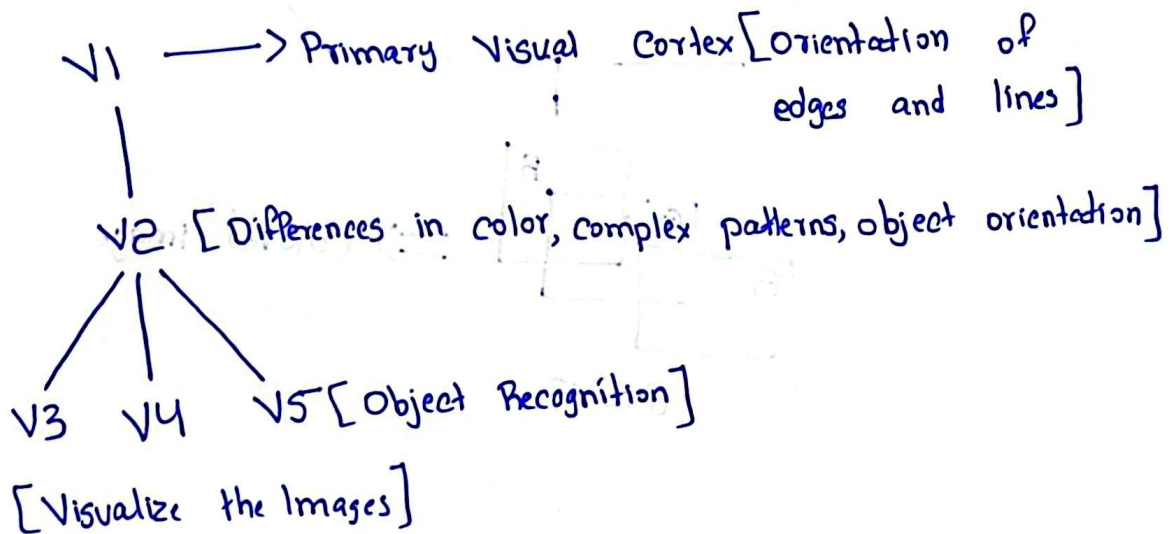
Object detection

Segmentation

## → Cerebral Cortex and Visual Cortex

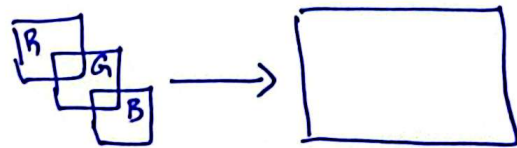
Visual Cortex (V1-V5)

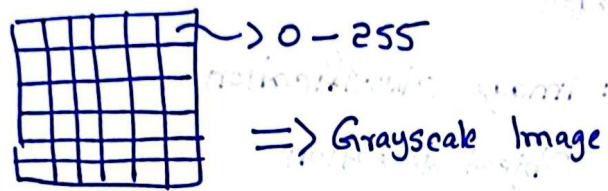
[Region of the brain that receives, integrates and processes visual information relayed from the retina]



## → RGB Images and GrayScale Images

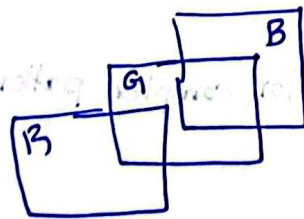
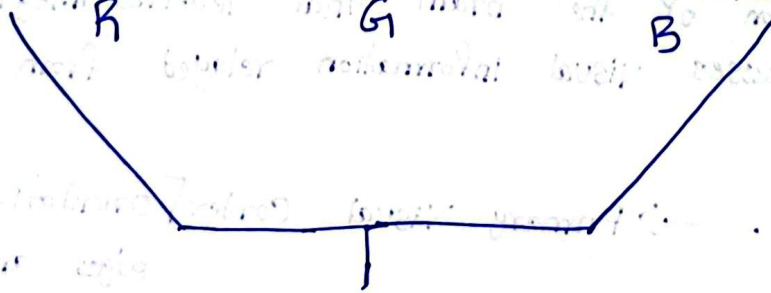
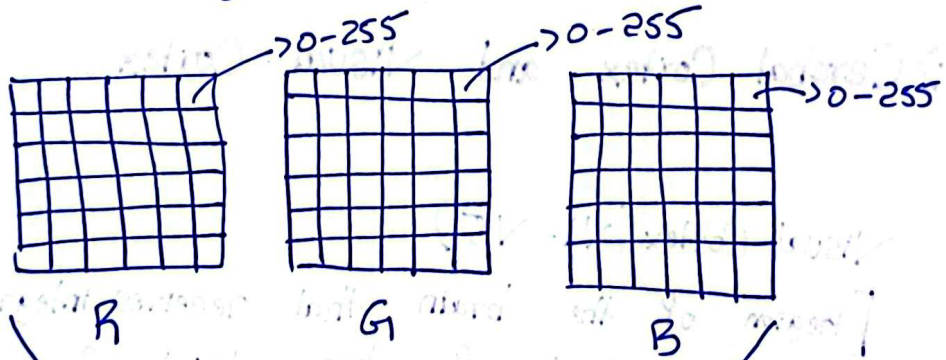
Image { Red ☐  
Green ☐  
Blue ☐





$6 \times 6 \times 1$

Grayscale



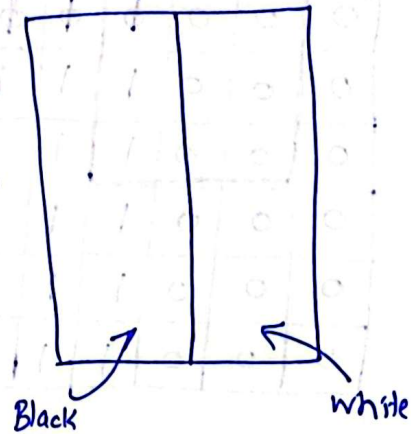
$6 \times 6 \times 3$

$\Rightarrow$  Colored Image

# → Convolution Operations in CNN

0	0	0	255	255	255
0	0	0	255	255	255
0	0	0	255	255	255
0	0	0	255	255	255
0	0	0	255	255	255
0	0	0	255	255	255

6x6x1



## Steps

### ① Normalize

↳ convert pixels between 0 and 1  
(divide by 255)

0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1

### ② Apply Filters

0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1

6x6x1

x

+1	0	-1
+2	0	-2
+1	0	-1

3x3  
Filter

=

0	-4	-4	0
0	-4	-4	0
0	-4	-4	0
0	-4	-4	0

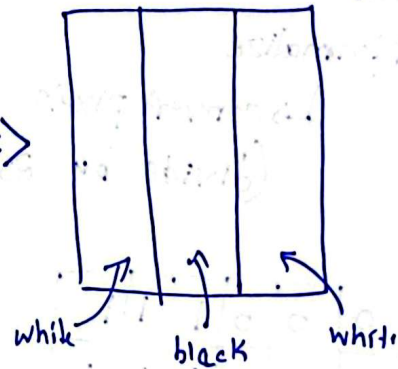
### ③ Stride

0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1

Stride = 1

### ④ Denormalize the output

255	0	0	255
255	0	0	255
255	0	0	255
255	0	0	255



### > Filters

+1	0	-1
+2	0	-2
+1	0	-1

vertical edge filter

+1	+2	+1
0	0	0
-1	-2	-1

horizontal edge filter

In real world, the values and size of filters is not fixed.

### > Formula

$$\begin{aligned}
 & \text{no. of filters} \rightarrow \\
 & \text{no. of pixels} - \text{filters} + 1 \\
 & = n - f + 1 \\
 & = 6 - 3 + 1 \\
 & = 4
 \end{aligned}$$



## → Padding in CNN

0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1

↓ Padding

	0	0	0	1	1	1
	0	0	0	1	1	1
	0	0	0	1	1	1
	0	0	0	1	1	1
	0	0	0	1	1	1
	0	0	0	1	1	1

8x8

x

+1	0	-1
+2	0	-2
+1	0	-1

=

	0	-4	-4	0	
	0	-4	-4	0	
	0	-4	-4	0	
	0	-4	-4	0	

6x6

$$n - f + 2p + 1$$

$$6 - 3 + 2p + 1 = 6$$

$$2p + 1 + 3 = 6$$

$$2p = 6 - 4$$

$$p = 2/2$$

$$p = 1$$

## > Types of Padding

① Zero Padding: fill padding values by 0.

② Neighbor Padding: fill padding by neighbor values

## → Pooling

- ① Min Pooling
- ② Max Pooling
- ③ Average Pooling

-4	0	-2	4	1
3	1	0	2	1
1	0	1	1	1
4	6	5	1	0
-1	2	0	0	0

Pool-size = 3  
Stride = 2

3	4
6	5

Max  
Pooling

-4	-2
-1	0

Min  
Pooling

0	1
2	1

Average  
Pooling